

Mininet-WiFi: Emulação de Redes Sem Fio Definidas por Software com suporte a Mobilidade

Ramon dos Reis Fontes¹, Christian Esteve Rothenberg¹

¹Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
Caixa Postal 6101, 13083-970 – Campinas – SP – Brazil

{ramonrf, chesteve}@dca.fee.unicamp.br

Abstract. *Software-Defined Wireless Networks is being regarded as a paradigm shift in network architectures based on a clear and programmatic between control and data planes enabling the control plane innovation and the underlying infrastructure to be abstracted from applications and network services. Despite ongoing efforts to realize the vision of SDWN, many questions remain open from multiple perspectives towards their feasibility, including effective means to experiment and rapid prototype candidate applicable to the real world. To this end, we propose the emulation of wireless networks with the use of Mininet-WiFi, as a potential approach to leverage research on SDWN by using existing protocol and application stacks while mimicking the behavior of real wireless networks.*

Resumo. *As redes sem fio definidas por software (SDWN) têm sido relacionadas na mudança de paradigma em arquiteturas de redes baseado na programática entre os plano de controle e dados, permitindo a inovação no plano de controle e a utilização da infraestrutura subjacente por aplicações e serviços de rede. Apesar dos esforços na construção de uma visão sobre SDWN muitas questões ainda permanecem abertas em relação a sua viabilidade, incluindo meios efetivos para a experimentação e prototipação aplicáveis no mundo real. Sendo assim, nós propomos a emulação das redes sem fio através do Mininet-WiFi como alternativa para alavancar pesquisas em SDWN através da utilização de pilhas de protocolos e aplicações existentes ao mesmo tempo em que é capaz de reproduzir o comportamento de redes sem fio reais.*

1. Introdução

As redes sem fio consistem em um número de nós que comunicam entre si através de um canal sem fio com a vantagem da mobilidade [Holt and Huang 2010], entre outras coisas. O padrão para redes sem fio mais amplamente aceito publicado pelo Instituto Europeu de Normas de Telecomunicações (ETSI) é o IEEE 802.11, principalmente porque muitos dispositivos portáteis possuem chipsets 802.11 incorporados, além de equipamentos como pontos de acessos serem relativamente baratos.

Embora sejam muito importantes, ainda existem barreiras estruturais que impedem a inovação em redes sem fio. Além disso, as grandes infraestruturas sem fio não estão completamente acessíveis, pois existem restrições no seu uso ou exigem autenticação [Yap et al. 2010]. A questão não é abrir completa e gratuitamente o acesso

às redes sem fio, mas permitir que usuários se conectem a várias redes (preservando requisitos de segurança e de qualidade), abrindo enorme capacidade de cobertura, além de possibilitar a inovação contínua, conforme proposto em [Santos et al. 2013].

A fim de quebrar com as barreiras e caminhar rumo à inovação, propomos neste trabalho a utilização do Mininet-WiFi, emulador que permite a adoção do conceito de redes definidas por software (SDN) [Kreutz et al. 2014, McKeown et al. 2008] para redes sem fio, mais conhecido por Software Defined Wireless Networks (SDWN) [Jagadeesan 2014] [Costanzo et al. 2012] nas pesquisas e experimentações em SDWN. Através de um computador o Mininet-WiFi permite emular diferentes tipos de cenários como redes *infra-estruturadas*, *ad hoc*, *mesh*, além de permitir a mobilidade dos nós, suportando atualmente até a integração com redes veiculares. Alguns destes cenários, além de outros, estão disponíveis em um canal na Internet¹.

A emulação de rede já tem sido amplamente utilizada na avaliação de desempenho, testes e depuração de protocolos e também em pesquisas sobre redes de computadores [Fall 1999], sendo uma abordagem muito importante para o desenvolvimento de novas soluções e avaliações das atualmente existentes. Emuladores possibilitam aos usuários terem controle total sobre suas experimentações e o Mininet-WiFi, mais especificamente, permite a utilização de qualquer aplicação de rede com pouca ou nenhuma alteração no *modus operandi* dessas aplicações.

A emulação para redes sem fio é mais desafiadora, uma vez que envolve muitos fatores relacionados com o meio, como autenticação, autorização e conta, políticas, interferências e gestão da mobilidade, técnicas de seleção de canal [Schulz-Zander et al. 2014], entre outros. Alguns dos requisitos fundamentais dessas plataformas experimentais é o suporte à mobilidade e propagação do meio. Para tais fins se faz necessário utilizar modelos capazes de representar com fidelidade o movimento de terminais móveis, pessoas, etc., além do comportamento esperado na comunicação como largura de banda, atraso, latência e perda de pacotes, da mesma forma como ocorre no mundo real.

Além desta introdução, os demais assuntos tratados neste trabalho estão organizados da seguinte forma: na seção 2 contextualizamos as redes sem fio definidas por software, também discutindo sobre os protocolos OpenFlow e CAPWAP; a seção 3 apresenta o emulador Mininet-WiFi; na seção 4 apresentamos quais serão as demonstrações a serem apresentadas no salão de ferramentas; na seção 5 são apresentadas algumas limitações e também algumas considerações para trabalhos futuros; e, finalmente, na seção 5 são apresentadas as conclusões deste trabalho.

2. Redes sem Fio Definidas por Software

As duas principais características das redes sem fio definidas por software ou SDWN (Software Defined Wireless Networks) [Jagadeesan 2014] [Costanzo et al. 2012] são o controle centralizado da rede através de software e a separação entre o plano de controle (parte inteligente da rede) e o plano de dados (responsável pelo encaminhamento de pacotes). Essa é uma abordagem já adotada pelas redes definidas por software ou SDN (Software-Defined Networking) [Kreutz et al. 2014], que permite administradores

¹<https://goo.gl/kPslqz>

de rede especificarem o comportamento da rede de forma centralizada através de uma plataforma de controle, sendo o protocolo OpenFlow [McKeown et al. 2008] o mais utilizado pela comunidade acadêmica e também pela indústria.

SDWN tem sido tema de algumas pesquisas relacionadas a SDN, incluindo crescente atenção por parte das operadoras de redes móveis [Bernardos et al. 2014, Sama et al. 2015] e sua sinergia com as funções de redes virtualizadas - NFV (Network Function Virtualisation) [Han et al. 2015]. Contudo, embora a especificação 1.5 do protocolo OpenFlow esclareça como deve ser o tratamento de fluxos que entram e saiam por uma mesma porta, caso comum em redes sem fio, a atual especificação do protocolo não traz soluções para assuntos relacionados a redes wi-fi, incluindo os desafios apresentados anteriormente acerca da emulação de redes sem fio.

A separação do plano de dados e controle não é uma novidade para redes sem fio. A IETF já havia padronizado o LWAPP (*Lightweight Access Point Protocol*) - RFC541 e CAPWAP (*Control And Provisioning of Wireless Access Points*) - RFC4564 alguns anos atrás e muitas empresas utilizam esses protocolos na gestão de suas redes sem fio. O LWAPP é um protocolo que define as mensagens de controle para configuração, autenticação e operações em tempo real, enquanto que o CAPWAP é um protocolo interoperável baseado no LWAPP que permite que controladores administrem pontos de acesso de forma centralizada.

Existem uma série de trabalhos publicados que propõem soluções para experimentação em SDWN, a exemplo de: *OpenRoads* [Yap et al. 2010], *Odin* [Suresh et al. 2012], *OpenRF* [Kumar et al. 2013], *Ethanol* [Moura et al. 2015]. Outros trabalhos como *CloudMac* [Dely et al. 2012] e *Chandelle*² mencionam CAPWAP em suas soluções. Para o *CloudMac* protocolos de gerenciamento de redes sem fio como o CAPWAP são difíceis de serem estendidos e oferecer novas funcionalidades e por isso propõe uma arquitetura para a gestão de redes sem fio. O *Chandelle*, por sua vez, propõe uma solução de *roaming* para redes wi-fi de forma rápida e suave com a utilização do protocolo OpenFlow, mas também apresenta um problema na integração de comutadores tradicionais e o CAPWAP mostrando que as redes tradicionais não estão prontas para *roaming*.

Ainda hoje a única forma realística para experimentar wi-fi e OpenFlow é através da utilização de *firmware* de código-fonte aberto e soluções como o *OpenWRT*, que permite transformar simples roteadores sem fio em comutadores OpenFlow. Contudo, mesmo o *OpenWRT* possui limitações de recursos, escalabilidade, controle e reproduzibilidade, uma vez que estamos tratando com dispositivos físicos e não são todos eles que são capazes de permitir a execução da *firmware* com o protocolo OpenFlow. O Mininet-WiFi, por outro lado, é excelente quando se pretende trabalhar nestes tipos de adversidades, pois possui escalabilidade razoável, permitindo pesquisas em SDWN com a vantagem de uma rápida prototipagem e benefícios já conhecidos pela grande comunidade do Mininet [Lantz et al. 2010]. Além disso, levando em consideração que o plano de controle de redes sem fio é mais complexo que em redes cabeadas, certamente os ganhos obtidos também serão maiores.

²<http://www.usenix.org/sites/default/files/ons2014-poster-monin.pdf>

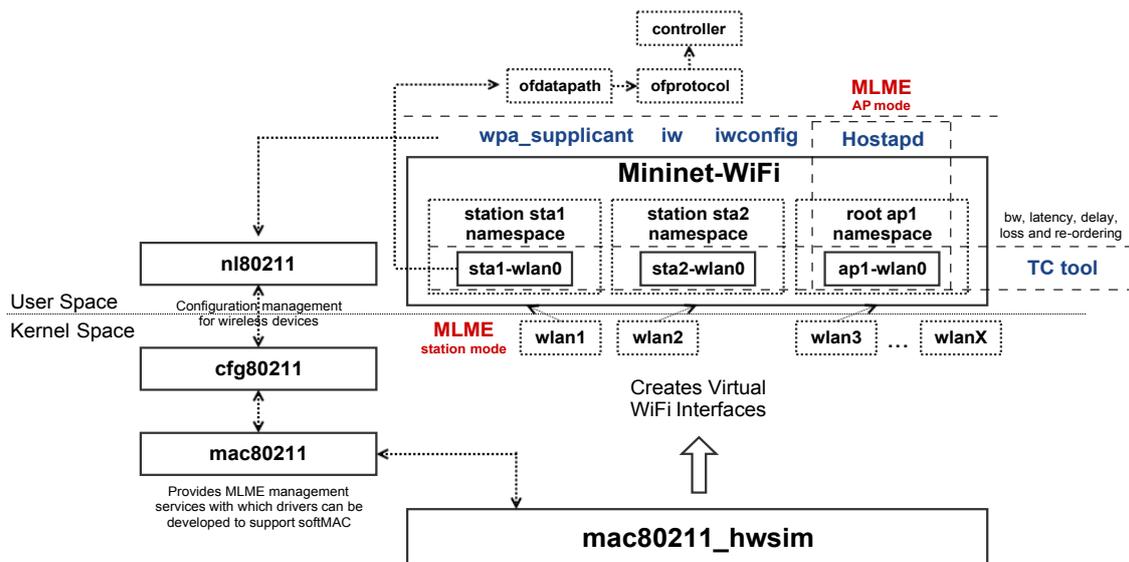


Figura 1. Componentes para funcionamento do Mininet-WiFi.

3. Mininet-WiFi

O Mininet-WiFi [Fontes et al. 2015] é um projeto de código-fonte aberto³, desenvolvido na linguagem de programação Python e que já conta com a colaboração de vários interessados da comunidade acadêmica que estão atuando em pesquisas relacionadas a SDWN. Estações e pontos de acesso são virtualizados e os dispositivos tradicionais já presentes no Mininet [Lantz et al. 2010] também podem ser utilizados, como computadores, comutadores e controladores OpenFlow. Uma vez permitindo a utilização de controladores OpenFlow, o Mininet-WiFi possibilita o tratamento de fluxos de pacotes utilizando o protocolo OpenFlow [McKeown et al. 2008] da mesma forma como já é realizado no Mininet para experimentos em redes definidas por software (SDN) [Yang et al. 2014].

Baseado no Mininet, o principal emulador para redes OpenFlow e SDN e também no *driver* wi-fi mais utilizado em sistemas Linux, o *SoftMac*, o principal objetivo do Mininet-WiFi é explorar as características das redes wi-fi e colaborar com o avanço em *Software-Defined Wireless Networking*. Os principais componentes envolvidos no funcionamento deste emulador estão apresentados na Figura 1. No espaço do Kernel, o módulo *mac80211_hwsim* é responsável por criar as interfaces wi-fi virtuais necessárias para a criação das estações e pontos de acessos. Ainda no espaço do Kernel o MLME⁴ (*Media Access Control Sublayer Management Entity*) é realizado nas estações, enquanto no espaço de usuário o *Hostapd* é o responsável por realizar essa tarefa. O *Hostapd* é o software que faz com que uma placa de rede wi-fi virtual comporte-se como uma interface de ponto de acesso.

No Mininet-WiFi também são utilizados os utilitários *iw*, *iwconfig* e o *wpa_supplicant*. Os dois primeiros são utilizados para suporte na configuração e obtenção de informações das interfaces sem fio e o último é utilizado juntamente com o *Hostapd*,

³<https://github.com/intrig-unicamp/mininet-WiFi>

⁴Algumas das funções realizadas pelo MLME são a autenticação, associação, envio e recebimento de *beacons*, etc.

dentre outras coisas, para o suporte ao WPA (*Wi-Fi Protected Access*). Além deles, uma outra ferramenta fundamental para o funcionamento do Mininet-WiFi é o *TC (Traffic Control)*. O *TC* é responsável por fazer o controle das interfaces virtuais, fazendo com que diferentes valores para largura de banda, atraso, latência e perda de pacotes sejam atribuídos a elas, permitindo representar com maior fidelidade o comportamento de troca de dados do mundo real.

Para instalar o Mininet-WiFi basta seguir as 4 (quatro) etapas que estão descritas na página do código-fonte. O principal requisito para instalar e executá-lo é o sistema operacional Linux, em especial distribuições Ubuntu a partir da versão 14.04. É possível que ele funcione em outros sistemas operacionais Linux, contudo, ainda não foram realizados testes nesse sentido. Também não há uma documentação particular ao Mininet-WiFi, porém, como ele é baseado no Mininet ele incorpora toda a arquitetura do Mininet, acrescido dos recursos wi-fi apresentados na Figura 1. Além disso, os usuários contam com uma comunidade ativa⁵ onde muitas questões podem ser discutidas e esclarecidas.

Uma introdução ao emulador Mininet-WiFi foi previamente publicada em [Fontes et al. 2015], no entanto, o emulador está em contínuo desenvolvimento e novos recursos estão sendo disponibilizados à medida que são implementados. Alguns desses recursos são: (i) a mobilidade, onde têm sido aprimorada nas novas versões através da implementação de novos modelos de mobilidade; (ii) propagação do sinal, onde modelos de propagação conceituados têm sido implementados, como o *Friis Propagation Loss Model* (também conhecido como *Free Space Path Loss*) e o *Two-Ray Ground Propagation Loss Model* [Rappaport 2001]; (iii) roteamento para redes mesh, com a possibilidade de utilizar um roteamento personalizado; (iv) integração com redes veiculares, onde já existe suporte para um simulador veicular bastante conhecido pela comunidade acadêmica, o SUMO⁶; entre outros.

4. Demonstração

Para demonstrar o funcionamento do Mininet-WiFi, apresentaremos quatro casos de uso destacando as evoluções desde a publicação anterior. O primeiro caso diz respeito a criação de uma simples topologia com duas estações e um ponto de acesso. No segundo caso utilizaremos uma topologia com pontos de acesso, comutadores, estações, computadores e controlador OpenFlow. No terceiro caso, mostraremos um cenário de mobilidade, onde é possível, dentre outras coisas, verificar o conceito de *handover*. E no último e quarto caso, permitiremos que usuários presentes na demonstração possam se conectar através de seus dispositivos móveis à uma rede emulada. Alguns comandos internos recentemente desenvolvidos e apresentados na página wiki⁷ também poderão ser utilizados durante a demonstração, além do recurso gráfico oferecido pelo emulador (Figura 2).

No **caso 1**, será demonstrado como é realizada a comunicação entre duas estações através do comando `ping`, além de verificar a conectividade entre estações e pontos de acesso através do comando `iwconfig`. No **caso 2**, mostraremos que a comunicação entre estações e computadores é realizada com êxito quando um controlador interno em modo reativo é utilizado. Em seguida definiremos um controlador externo inexistente

⁵<https://goo.gl/rRQX42>

⁶http://sumo.dlr.de/wiki/Main_Page

⁷<https://github.com/intrig-unicamp/mininet-wifi/wiki>

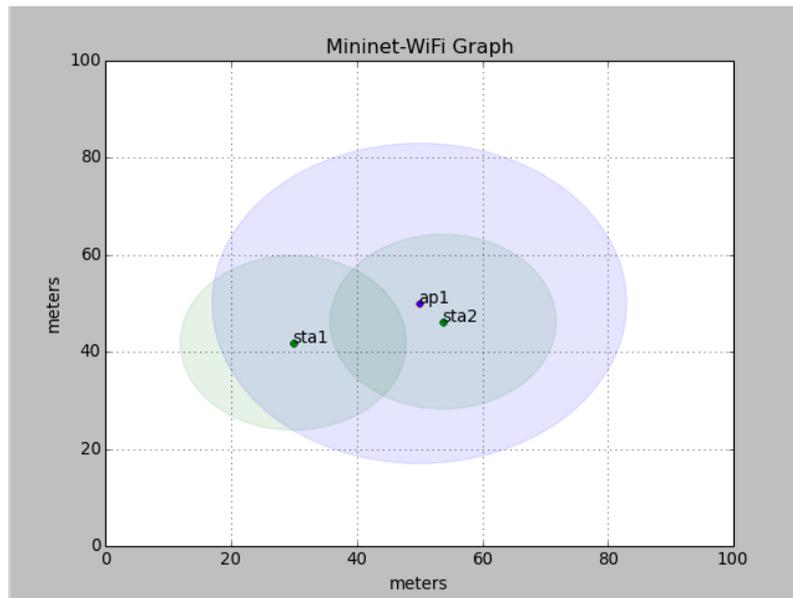


Figura 2. Mininet-WiFi em execução.

com o intuito de mostrar a importância que o controlador OpenFlow têm na comunicação entre os componentes de rede. No **caso 3**, além de verificar o conceito de *handover*, também mostraremos o impacto que a distância pode causar na comunicação entre duas estações em movimento. E finalmente, no **caso 4**, iremos criar uma topologia em ambiente emulado e qualquer usuário presente na demonstração poderá utilizar seus equipamentos móveis de forma a interagir com o ambiente emulado. Além destes casos, uma série de outros estão disponíveis em um canal⁸ na Internet e no diretório *examples* (localizado no diretório raiz de instalação do Mininet-WiFi).

5. Limitações e Trabalhos Futuros

Algumas limitações têm sido percebidas durante o desenvolvimento do Mininet-WiFi, a exemplo do módulo responsável pela criação das placas de redes virtuais, o *mac80211_hwsim*, que limita a 100 (cem) o número de placas de redes que podem ser criadas numa única topologia. Por outro lado, é possível recompilar o módulo apenas alterando uma limitação imposta pela função *init_mac80211_hwsim*. Não há nenhuma referência que justifique esse limite e por isso não sabemos explicar o motivo desta limitação. Em testes realizados com o módulo recompilado conseguimos criar placas virtuais superior ao limite imposto pela função sem demais problemas.

Com relação à propagação no meio ainda existem limitações que devem ser consideradas como, por exemplo, responder com maior exatidão as interferências de sinal e taxa de transferência de dados coerente com o encontrado em um ambiente real. Estas limitações ainda serão tratadas e por esse motivo também são consideradas como trabalhos futuros. A solução para elas deve começar pela utilização de modelos de propagação já consagrados (além dos atualmente suportados) e bastante conhecidos pela comunidade acadêmica, como o *Log Normal Shadowing* [Rappaport 2001], *Nakagami* [Nakagami 1960], *Okumara-hara* [Hata 1980], dentre outros.

⁸<https://goo.gl/kPslqz>

6. Conclusões

Neste trabalho apresentamos a ferramenta Mininet-WiFi, um emulador para redes sem fio definidas por software (SDWN). Devido ao sucesso e sua ampla adoção pela comunidade de pesquisa, o protocolo OpenFlow é apresentado como alternativa para SDWN. Por não haver ainda definições sobre o seu funcionamento em redes sem fio, acreditamos que o emulador Mininet-WiFi pode trazer uma série de ganhos, sendo objeto de suporte em pesquisas sobre SDWN com a utilização do protocolo OpenFlow, CAPWAP e outros protocolos que venham a surgir nesse segmento, por exemplo no contexto de virtualização de funções de rede (NFV - Network Function Virtualization).

Como foi visto, embora a separação do plano de dados e controle não seja uma novidade em redes sem fio, ainda não há uma definição clara sobre como deve ser o seu funcionamento, faltando plataformas de referência que forneçam suporte para o desenvolvimento de novas pesquisas. Sendo assim, propomos a utilização do emulador Mininet-WiFi, onde demonstramos alguns dos recursos atualmente suportados através de alguns casos de uso, destacando as evoluções que o emulador teve desde a sua publicação anterior. Discutimos também algumas limitações da implementação atual e considerações de resolução dessas limitações em futuras versões do emulador.

Agradecimentos

Trabalho parcialmente financiado pelo processo nº 2014/18482-4, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP).

Referências

- Bernardos, C., La Oliva, A., Serrano, P., Banchs, A., Contreras, L. M., Jin, H., and Zuniga, J. C. (2014). An architecture for software defined wireless networking. *Wireless Communications, IEEE*, 21(3):52–61.
- Costanzo, S., Galluccio, L., Morabito, G., and Palazzo, S. (2012). Software Defined Wireless Networks: Unbridling SDNs. In *Software Defined Networking (EWSDN), 2012 European Workshop on*, pages 1–6. IEEE.
- Dely, P., Vestin, J., Kessler, A., Bayer, N., Einsiedler, H., and Peylo, C. (2012). Cloud-MAC - An OpenFlow based architecture for 802.11 MAC layer processing in the cloud. In *Globecom Workshops (GC Wkshps), 2012 IEEE*, pages 186–191. IEEE.
- Fall, K. (1999). Network emulation in the vint/ns simulator. In *Proceedings of the fourth IEEE Symposium on Computers and Communications*, pages 244–250.
- Fontes, R., Afzal, S., Brito, S., Santos, M., and Esteve Rothenberg, C. (2015). Mininet-WiFi: emulating Software-Defined wireless networks. In *2nd International Workshop on Management of SDN and NFV Systems, 2015*, Barcelona, Spain.
- Han, B., Gopalakrishnan, V., Ji, L., and Lee, S. (2015). Network function virtualization: Challenges and opportunities for innovations. *Communications Magazine, IEEE*, 53(2):90–97.
- Hata, M. (1980). Empirical formula for propagation loss in land mobile radio services. *Vehicular Technology, IEEE Transactions on*, 29(3):317–325.

- Holt, A. and Huang, C.-Y. (2010). *802.11 Wireless Networks: Security and Analysis*. Springer Publishing Company, Incorporated, 1st edition.
- Jagadeesan, Nachikethas A., K. B. (2014). Software-defined networking paradigms in wireless networks: A survey. *ACM Comput. Surv.*, 47(2):27:1–27:11.
- Kreutz, D., Ramos, F. M. V., Veríssimo, P., Rothenberg, C. E., Azodolmolky, S., and Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *CoRR*, abs/1406.0440.
- Kumar, S., Cifuentes, D., Gollakota, S., and Katabi, D. (2013). Bringing cross-layer mimo to today’s wireless lans. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM ’13, pages 387–398, New York, NY, USA. ACM.
- Lantz, B., Heller, B., and McKeown, N. (2010). A network in a laptop: Rapid prototyping for software-defined networks. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, Hotnets-IX, pages 19:1–19:6, New York, NY, USA. ACM.
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., and Turner, J. (2008). Openflow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2):69–74.
- Moura, H., Bessa, G. V. C., Vieira, M. A. M., and Macedo, D. F. (2015). Ethanol: Software defined networking for 802.11 wireless networks. In *IFIP/IEEE International Symposium on Integrated Network Management, IM 2015, Ottawa, ON, Canada, 11-15 May, 2015*, pages 388–396.
- Nakagami, M. (1960). The m-distribution – A general formula of intensity distribution of rapid fading. In Hoffmann, W. C., editor, *Statistical Methods in Radio Wave Propagation*. Elmsford, NY.
- Rappaport, T. (2001). *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2nd edition.
- Sama, M. R., Contreras, L. M., Kaippallimalil, J., Akiyoshi, I., Qian, H., and Ni, H. (2015). Software-defined control of the virtualized mobile packet core. *IEEE Communications Magazine*, 53(2):107–115.
- Santos, M. A. S., de Oliveira, B. T., Margi, C. B., Nunes, B. A. A., Turetletti, T., and Obraczka, K. (2013). Software-defined networking based capacity sharing in hybrid networks. In *ICNP*, pages 1–6.
- Schulz-Zander, J., Suresh, L., Sarrar, N., Feldmann, A., Hühn, T., and Merz, R. (2014). Programmatic Orchestration of WiFi Networks. In *USENIX ATC 14*, pages 347–358, Philadelphia, PA. USENIX Association.
- Suresh, L., Schulz-Zander, J., Merz, R., Feldmann, A., and Vazao, T. (2012). Towards programmable enterprise wlans with odin. In *Proceedings of HotSDN ’12*, pages 115–120, New York, NY, USA. ACM.
- Yang, M., Li, Y., Jin, D., Zeng, L., Wu, X., and Vasilakos, A. V. (2014). Software-defined and virtualized future mobile and wireless networks: A survey. *CoRR*, abs/1409.0079.
- Yap, K.-K., Sherwood, R., Kobayashi, M., Huang, T.-Y., Chan, M., Handigol, N., McKeown, N., and Parulkar, G. (2010). Blueprint for introducing innovation into wireless mobile networks. In *VISA ’10*, pages 25–32, New York, NY, USA. ACM.